12404

Postel

Network Information Center # 12404
Request for Comments # 412

G. Hicks
Utah
27-NOV-72
Rvcd 21 Dec 72

User FTP Documentation

The attached document is the HELP file for the Utah-10 implementation of the User FTP Process. This is what the user has typed on his console when he types 'HELP<cr>', and as such is the only documentation produced to date.

GPH

AKB

FTP User Documentation
November 27, 1972

I. Control Characters

There are several control characters that have meaning to the FTP process. First, the abort character is <control-z> (SUB). Next for editing, only <control-r>, <control-a>, <control-h>, <control-v>, <control-w>, <alt-mode>, <eol>, <space>, <control-x> and <rub-out> have any meaning. <control-a> (SOH) and <control-h> (BS) are the character delete keys. <control-w> is the word delete character. NOTE: When doing the ′ function, this character will delete the entire line typed. <control-v> says take the next character literally. <alt-mode> (ESC) terminates the command giving helpful noise words. <eol> and <space> are terminators also. These last two will give no command completion or noise words. All three echo as a space however. <control-x> (CAN) and <rub-out> (DEL) are the command abort characters. <control-r> (DC2) will retype the line as the command interpreter sees it. When using the ? as a prompter, for initial commands, it will type-out all commands that begin with that particular character string. If it is typed in a sub-command field, it will type out the prompt message only if it is the first character typed. At all other times, it will be accepted as part of the typed string.

II. General Information

The FTP user process is designed to make transferring files from one host on the ARPANET to another much easier than it has been up until now. To this end, the command language was designed to be as natural as possible.

The command interpreter is, of course, part of this idea. Therefore, to help the user as much as possible, there are several ways to find out what is expected in the form of commands, file-names, user-names, etc. When the user has typed nothing, typing a ? will cause the interpreter to type out all the commands that are available to the user. If the user has typed anything at all, then it will respond with all commands that begin with the particular character string.

So that the command language was as easy to learn as possible, the command interpreter will ′see′ nothing that is not part of a legal command. If the user types anything that is not expected, the character is not echoed and a bell is echoed instead.

III. Brief Command Listing

The commands that are expected and their syntax are listed below. NOTE: UPPER and lower case letters are identical.

<host-name>
D<decimal host number>
<octal host number>
;<any string> (useful for comments to a person that is linked to you)
MODE <mode name>
RETREIVE <remote file> (to) <local file>
GET <remote file> (to) <local file>
SEND <local file> (to) <remote file>
STORE <local file> (to) <remote file>
APPEND <local file> (to) <remote file>
RENAME <remote file> (to be) <new remote file name>
DELETE <remote file>
BYE
COPY <direction descriptor> <file group> (to) <file group> (see detailed description below)
LOGOUT
DDT
LIST <file group> (to file) <local file name>
SOCKET (NOT IMPLEMENTED YET)
ALLOCATE <decimal number of bytes>
QUIT
' (quote mark) <any string> (is sent directly to the foreign host)
HOST <any valid ARPANET host name or number>
USER <remote user name>
PASSWORD <password for remote user's name>
ACCOUNT <string or number for remote user's name>
STATUS (see description of status command below)
LOGIN <user name> <password> <optional account> <to host>
BYTE <decimal byte size>
TYPE <type descriptor>
STRUCTURE <structure descriptor>
TENEX
ASCII
VERBOSE

17. Detailed Description of the Commands

The commands and their syntax are described in greater detail below. The words in parenthesis are noise words. NOTE: upper and lower case are identical. Unless otherwise noted in the command description, all commands described are implemented.

<host name> or <host number>
Performs the ICP to the indicated host with explanatory remarks if the ICP is not possible.

MODE <mode descriptor>
Sets the mode of the data transfer connection according to the following sub-commands:
STREAM: bit stream, end of file is indicated by the data connection closing
BLOCK: formats (or expects formatted) data into blocks.
TEXT: Sends or retreives text. Forces TYPE ASCII and BYTE 8. This command sends the appropiate commands to the foreign host, then sends the data using the TELNET codes for EOR and EOF as per the FTP Protocol.
HASP: compresses data.
NOTE: Of the above sub-commands, only MODE STREAM and MODE TEXT are implemented at present.

RETREIVE (file) <remote file> (to file) <local file name>
Sends the retr command to the remote server, sets up the data connection according to any previous MODE, TYPE, BYTE commands. Puts the data coming on the data connection into the local file specified.

GET (file) <remote file name> (to file) <local file name>
See description of RETREIVE.

STORE (local file) <file name> (onto file) <remote file>
Accepts a local file name, does the formatting according to any previous BYTE, TYPE, MODE commands and sends it to the foreign host.

SEND (local file) <file name> (onto file) <remote file>
See description of STORE.

APPEND (local file) <file name> (to remote file) <remote file>

Does the same as a store except that the file is appended to the remote file rather that just writing over the file.

RENAME (existing file) <file name> (to be) <new file name>
Accepts the name of an old remote file and asks for a new file name, sends the appropiate commands and names to the foreign host causing the old file name to be replaced by the new file name.

DELETE (file) <file name>
This command causes the remote file to be marked for deletion. It does require that the command be confirmed twice.

BYE
Takes no arguments. Causes the server to terminate the current session with the user. The program will return to the EXEC MODE when the command has been acknowledged by the remote server.

COPY
This command does a variety of things. First, it allows the user to describe a file group. EG: *.mac, *.sav, etc. The *'s may be for foreign files or local files according to the following sub-commands:

REMOTE (remote file group) <remote file grouping>: Causes the user FTP to ask the server for the files in <remote file grouping>. Then asks the user to specify where each file is to go. A typical sequence might look like the following:

!copy remote (remote file group) ftp.*
Please be patient. Getting remote file names.
Got them.
!copy (file) <HICKS>FTP.MAC;1 (to file) ftp.mac [New file]
[Messages and etc. follow with the above line being repeated for each file in the remote file group.]

LOCAL (local file grouping) <local file group>: Causes the user FTP to accept the file group specified, and send the file names to the server leaving off the <directory name> and version number of the local file. The user may not specify *'d devices or *'d directorys. A line for this might look like the following:
!copy local (local file group) ftp.*;*
[Confirm]

Next, if the user does not specify any *'s for either the remote file or the local file, this command is exactly like the RETREIVE or STORE command described earlier.

LOGOUT
Takes no arguments. See description of BYE command for more details.

RESET
Takes no arguments. Causes the user process to close all connections (if necessary), all files (if necessary), and reset the programs parameters to their defaults.

DDT
If DDT is loaded, just starts DDT. If DDT is not loaded, it will load DDT and then start it at its initial start-up

location. This command will cause an abnormal interrupt if DDT is not found.

LIST (file group) <remote file group> (to file) <local file>
Esentially causes the remote server to do a directory command. The default is *.*, and since the listing will come on the data connection, the user must specify a local file. A sample line might look like the following:
!list (file group) *.* (going to file) tty: [ok]

SOCKET <socket descriptor>
Will accept a socket descriptor for the data to go to or come from. At present it is NOT IMPLEMENTED because we have not decided on the format of the command to the server.

ALLOCATE <decimal number of bytes>
Accepts as its argument a decimal number of bytes (of the specified size) telling the server how many bytes of storage to reserve for the next store or append.

QUIT
Takes no arguments. Returns the console to the EXEC. The program may be continued with no harm done.

' [Quote mark]
This command allows the user to send arbitrary strings to the remote server. At present, when talking to a TENEX site, it is only useful for doing the 'MAIL' command. Other sites may have help commands or whatever.

HOST <ARPANET host name or number>
Allows the user to specify an ARPANET host without actually connecting to the host. This specified host will be the ont that is connected to (or the attempt will be made anyway) when the next 'immediate' command is executed. EG: LOGIN, RETREIVE, STORE, LIST, etc.

USER (name is) <remote user's name>
Sets the user name for the remote system. Useful for systems that require no password for the specified name. Does nothing until an 'immediate' command is executed. NOTE: For most of the commands the user must be 'logged into' the remote system. The program knows this and so will not let the user execute any commands (RETREIVE, STORE, APPEND, LIST, STATUS, RENAME, DELETE) until he has at least specified a remote user name. The LOGIN command does this as does this command.

STATUS (of the) <sub-cmd> (at host) <ARPANET host>
Accepts as its argument one of the following sub-commands:
SERVICE (at host) <ARPANET host name or number>: Attempts to perform the ICP to the specified host. Does not disturb any existing connections.
<ARPANET host name or number>: See description of STATUS

SERVICE above.
STATUS (of file) <remote file group>: Similar to the LIST command described above except that the listing does not come back on the data connection but on the TELNET connection. See LIST command for more details.
<COMMAND-TERMINATOR>: Just sends the STAT command. Typically, is good for finding out where you are and perhaps who you are.

PASSWORD (is) <password for remote user's name>
Sets the password for the remote user's name. It is NOT echoed and does nothing until the next 'immediate' command is executed.

ACCOUNT (is) <account number or string>
Accepts a string or number that can be charged for any activity the user specifies. EG: STORE, RETREIVE or APPEND. Useful only for systems that require this information of course. Also does nothing until the next immediate command is executed.

LOGIN <user name> <password> <optional account>
The format of this command is identical to the TENEX login command. It accepts a user name, password, and an optional account number. If no host has been specified, it will ask for the host to be connected to. This command will cause the ICP to take place if necessary.

BYTE (size is) <decimal byte size>
This command sets the byte size for the data connection. The default byte size is 8 bit bytes.

TYPE (is) <type descriptor>
Takes as its argument one of the following subcommands:
TENEX: Shorthand that sets TYPE IMAGE and BYTE 36.
EBCDIC: Says that the data will be Ebcdic.
PRINT: Says that the data is an Ascii print file with ASA vertical format controls.
LOCAL: Forces the user and server FTP to accept whatever byte size the user has specified. For the present, this type is treated the same as for IMAGE TYPE.
IMAGE: Does the same as LOCAL. Just forces the user and server FTP to accept the specified byte size without doing any translation as is done for ASCII type. For most efficent usage of this command, the matching byte size should be BYTE 36 when using TENEX sites as servers.
ASCII: Sets TYPE ASCII and 8 bit bytes. Useful for text files.
NOTE: Of the above sub-commands, only PRINT and EBCDIC are not implemented yet.

STRUCTURE (is) <file structure descriptor>
Accepts as its argument one of the following sub-commands:
RECORD: Says the data on the data connection has record structure information contained in it. NOTE: Since TENEX

does NOT support record structured files per se, this mode is only useful for sending text files to or retreiving text files from non-TENEX sites. The <EOL> in these files is converted to the TELNET EOR code for transmission and EOR to <EOL> etc. for reception.

FILE: Says the data stream has no record structure information in it.

TENEX
See description under TYPE TENEX.

ASCII
See description under TYPE ASCII.

VERBOSE
This command sets the command completion/no completion flag. The program assumes the NO VERBOSE mode and thus does not complete commands when <space> or <eol> is used as a command terminator. It also causes the noise words to be typed if in the verbose mode.

V. Sample Session using the FTP

The following scenario shows some uses of the FTP. Everything that the user types is underlined.

```
 UTAH-TENEX 1.29.03 EXEC 1.38
@LOGIN (USER) HICKS (PASSWORD) (ACCOUNT #) 1200
 JOB 24 ON TTY2 11-NOV-72 1:51
@TERMINAL (TYPE IS) 4
@ftp
FTP User Process 1.10 5-NOV-72.  Type Help<cr> for help.
!login (user) network (password) (account) 4 (to host) case-10
 CONNECTION IS OPEN TO CASE-10.
< CASE-10 FTP Server 1.14.0 - at SAT 11-NOV-72 2:00-EST
!tenex ? confirm with carriage return
 tenex
!copy remote (remote group) ftp.*
 Please be patient.  Getting remote file-names.
 Got them.
!copy (file) <NETWORK>FTP.MAC;32 (to file) ftp.mac [New file]
< IMAGE retreive of <NETWORK>FTP.MAC;32 started.
< Transfer completed.
!copy (file) <NETWORK>FTP.SAV;5 (to file) FTP.SAV;5 [New version]
< IMAGE retreive of <NETWORK>FTP.SAV;5 started.
< Transfer completed.
!copy (file) <NETWORK>FTP.REL;1 (to file) xxx
!copy (file) <NETWORK>FTP.HELP;5 (to file) ftp.help [New version]
 copy (file) <NETWORK>FTP.HELP;5 (to file) FTP.HELP;1
< IMAGE retreive of <NETWORK>FTP.HELP;5 started.
< Transfer completed.
Done...
!usc-isi
 Closing connections to CASE-10.
 CONNECTION IS OPEN TO USC-ISI.
< USC-ISI FTP Server 1.14.0 - at SAT 11-NOV-72 2:15-PST
!log
(user) hicks
(password)
(account) ? account number or string for remote user's name
 log
(user) hicks
(password)
(account) 1
!send (local file) ftp.sav;5 (to remote file) ftp.sav
< STORE of <HICKS>FTP.SAV;1;P777752;A1, IMAGE type, started.
< Transfer completed.
!co r acs.mac (to local file) acs.mac [New file]
< IMAGE Retreive of <HICKS>ACS.MAC;3 started.
< Transfer completed.
!copy 1 ? Local file group, *'s ok - TENEX sites only.
 copy 1 ftp.*
 [Confirm]
< Store of <HICKS>FTP.MAC;1;P777752;A1, IMAGE type, Started.
< Transfer completed.
< Store of <HICKS>FTP.SAV;2;P777753;A1, IMAGE type, Started.
```

```
< Transfer completed.
< Store of <HICKS>FTP.HELP;1;P777752;A1 IMAGE type, Started.
< Transfer completed.
!bye
!< BYE command recieved.
< Therefore, connection terminated.
@logout
[logout messages go here]
```